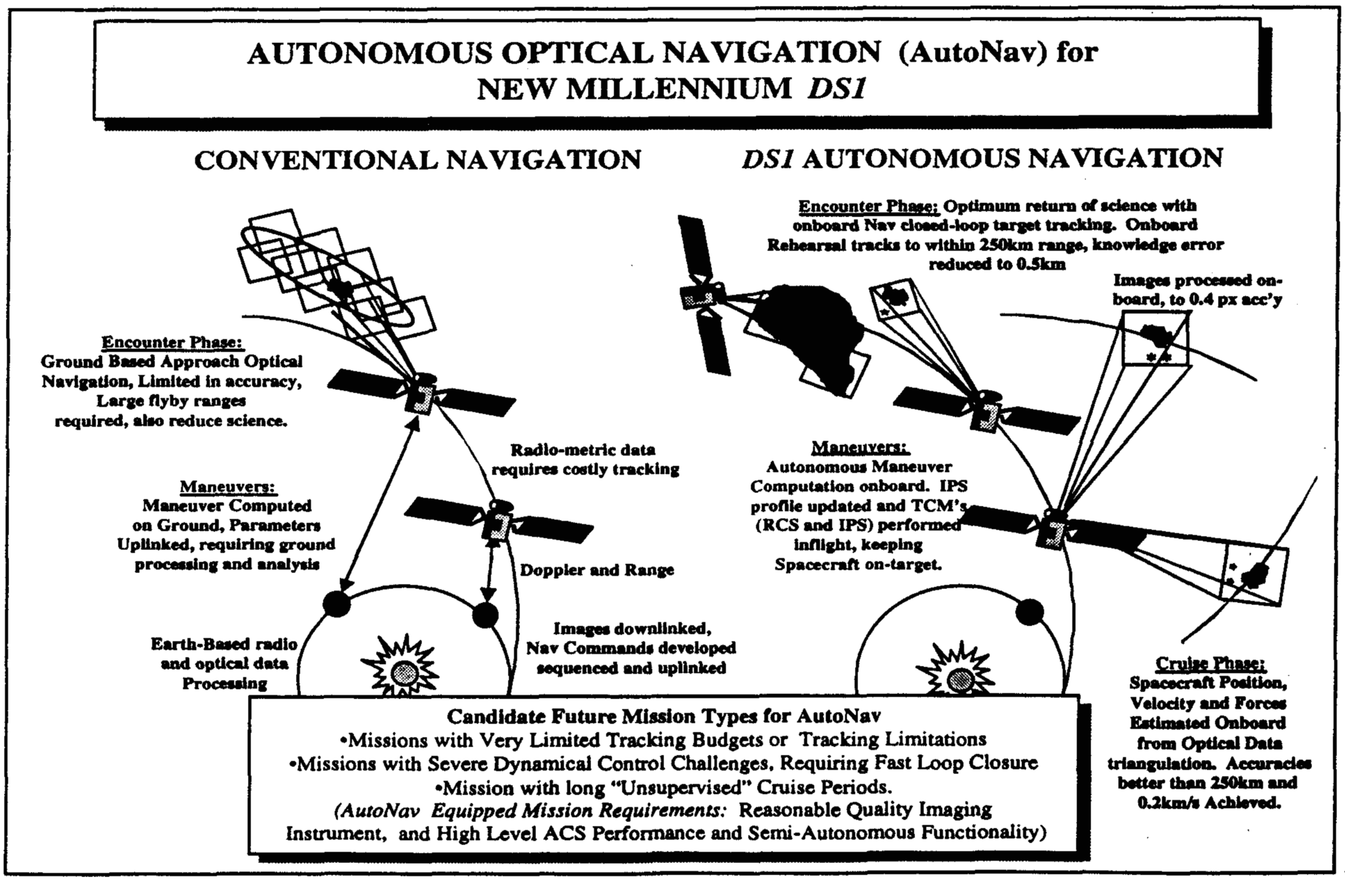
USING AUTONOMOUS NAVIGATION FOR INTERPLANETARY MISSIONS: MISSION OPERATIONS WITH *DEEP SPACE 1* AUTONAV

J. E. Riedel, S. Bhaskaran, S. Desai, D. Han, B. Kennedy, T. McElrath,
G. W. Null, M. Ryne, S. P. Synnott, T. C. Wang, R. A. Werner
Jet Propulsion Laboratory; California Institute of Technology; 4800 Oak Grove; Pasadena, CA. 91109

Abstract

The first mission of NASA's New Millennium Program, *Deep Space 1*, has, as one of its principal demonstration-technologies, the first autonomous optical navigation system to be used in deep space. The AutoNav system is a set of software elements that interact with the imaging, attitude control and ion-propulsion systems aboard *DS1* in order to accomplish optical data taking, orbit determination, trajectory correction maneuvers, ion propulsion system (IPS) control, and encounter operations. The validation of this system in the flight of *DS1* to Braille was very successful. Despite very substantial problems with the *DS1* camera, the AutoNav system was eventually able to determine the spacecraft heliocentric position to better than 200km and .2m/s, as determined by ground-based radio navigation, which for this low-thrust mission, was itself a new technology. As well as achieving this principal goal of high quality interplanetary cruise orbit determination, AutoNav successfully completed many complex and difficult operations. These operations include provision of astronomical ephemeris data to non-navigational systems onboard, planning and execution of hours-long picture-taking sessions, planning and execution of sessions of ion-engine function, planning and execution of trajectory correction maneuvers, and successful completion of encounter activities during the Braille approach rehearsal, and for part of the actual approach, delivering the spacecraft to as close as 2.5km of the desired impact-plane-position, and starting encounter sequences to within 5 seconds of the actual encounter-relative start time. By virtue of the AuotNav system, new streamlined methods of operations were developed that minimized ground planning and verification requirements for extensive complicated sequences. These sequences included those for navigation and calibration picture taking, IPS operations, both IPS and conventional TCMs, and encounter science activity.

Figure 1: Diagrammatic and Comparative Description of DS1 AutoNav



Introduction

AutoNav is an autonomous onboard optical navigation system. The basis of many of the methods and techniques used in AutoNav were developed during the *Voyager* "Grand Tour" Missions (Ref. 1). Further developments, including some specific algorithms used in AutoNav, were developed for the *Galileo* Mission to Jupiter (Ref. 2). Incorporation of these techniques, as well as the development of the overall design of AutoNav is described in Ref. 3. The AutoNav system is both a set of computational elements (e.g. image processing, orbit determination, maneuver planning) that largely operate in the background using available CPU cycles, and an active "Nav Executive" that will on occasion take command of the spacecraft to execute Nav and Nav-related activities involving image taking, turning, and propulsion events. It is the former elements that give AutoNav the means of determining course and course corrections, but it is the latter which gives AutoNav autonomy, and is a principal means by which substantial ease of operations were achieved over what would other have otherwise been a much more difficult mission.

Operations Methodology of *DS1* with AutoNav

The AutoNav system was designed to operate onboard a spacecraft with only minimal support from other autonomous systems. In the case of DS1, ACS provided the only onboard autonomous subsystem. This required AutoNav to perform onboard planning and scheduling of the component elements of extensive spacecraft events, such as picture taking sessions, operation of the IPS, and trajectory correction maneuvers (TCMs). Also, the design of AutoNav had to provide for the ability of the ground operations team to allocate all of the resources used by AutoNav, including spacecraft and camera time, compute resources, and onboard mass storage. In other words, AutoNav had to be autonomous, but could only work when invited to do so by the ground team through the onboard sequence.

In accomplishing its tasks, AutoNav had to work in an environment that was tightly constrained. Examples of some of these constraints include: Only certain parts of the sky were observable with the camera, or available for possible thrusting, due to illumination constraints on the body of the spacecraft and the requirement to keep the solar-panels tightly focused on-sun. Turning the spacecraft had to be accomplished in certain special planes, to avoid triggering "constraint-avoidance" action, which would have caused extraordinary fuel use, or possible spacecraft damage. With limited storage resources onboard, AutoNav had to balance the use of camera time, and mass storage, by monitoring the progress of the image-processing activity.

An important design feature of AutoNav was ease of use. Only a few very high-level commands, with few or no parameters, are required to accomplish long and very intense activity on board the spacecraft. Yet these commands provide for a very wide range of variable behavior in the system. This is accomplished in part with a suite of data files that are maintained as necessary by the ground and updated by the spacecraft in the normal process of autonomous navigation.

AutoNav is fully autonomous, but is autonomous only upon invitation of ground controllers. Most importantly, AutoNav will cause physical spacecraft activity or intense computational action only when invited to do so by the ground, allowing controllers to be fully aware beforehand when such activities will occur; however, the particulars of each of these events will likely not be completely predictable. For the three autonomous events that involve onboard-engineered sequences of turns, thrusting, or picture taking, the ground limits AutoNav to predetermined periods of time, allowing careful budgeting of onboard time, instrument and computational resources. Table 1 is a summary of the AutoNav commands, and a more thorough description of the principal and auxiliary commands follows.

Principal AutoNav Commands

Nav_Do_OD: Causes Nav to: 1) trim the optical data arc to the predetermined length, 2) trim the propulsion history (non-grav) file to a corresponding length, 3) compute data residuals and partials for all data points in the data arc, 4) estimate position, velocity and non-grav paramters for the spacecraft state for each segment of the arc, 5) repeat steps 3 and 4 iteratively until converged, 6) write these solutions on the solution file, and 7) integrate the current best estimated spacecraft state forward to a pre-specified time (usually about a month into the future and write this to the spacecraft ephemeris file).

Nav_Do_TCM: Causes Nav to perform a TCM, by 1) obtaing the pre-computed specifications for the next TCM from the Maneuver File, 2) checking that there is a TCM scheduled within a specified time (e.g. 1 hour), 3) querying ACS for the specifications of the turn to the attitude of the burn, 4) commanding ACS to perform the turn, 5) if the TCM is an IPS TCM, commanding IPS to thrust for the specified time, at the specified thrust, or if the TCM uses the RCS (hydrazine thrusters), commanding ACS to perform the specified impulsive delta-v, 6) if there is a component element to the TCM, performing steps 1-6 on this leg, and 7) commanding ACS to turn the spacecraft to the terminal attitude.

Nav_Man_Plan: Causes AutoNav to compute the propulsive plan for the next control opportunity in its onboard schedule (Maneuver file), if any. This may be an RCS or IPS TCM or an IPS Mission Burn. A) For a Mission burn, AutoNav will 1) propagate the last spacecraft state entry on the OD file to the B-plane, obtaining the current target miss vector, 2) starting with a fixed number of Mission burn segments, ManPlan will compute the partial derivatives of B-plane impact position and time with respect to burn angles of each segment and the duration of the final burn. 3) An estimate of the changes in the burn angle and last-segment-duration is made, 4) the estimated angle changes are checked for violations of pointing constraint. If a violation occurs, then that angle is reset to the constraint limit. 5) Using steps 1-4 an iteration is made, 6) if after a fixed limit of iterations, step 5 has not converged (i.e. targeting is not "close-enough") then Mission Burn segments are added to the set being updated, and steps 1-6 repeated, and 7) if the solution converges, then the Maneuver file is overwritten with the updated plan, otherwise, if there is no convergence, the Maneuver file is left unchanged. B) For a TCM, AutoNav will 1) propagate the last spacecraft state entry on the solution (OD) file to the epoch of the next maneuver, 2) from that epoch to the next encounter, the state and state partial derivatives are computed, 3) the required delta-v at the maneuver time is computed, 4) repeat steps 2 and 3 iteratively until converged, 5) determine, via interaction with ACS whether the desired burn direction violates spacecraft constraints, 6) if so, ask ACS to "vectorize" this TCM, i.e. decompose the desired – but constrained - delta-v direction into two allowed directions, and 7) via steps 2,3 and 4 compute the delta-v associated with each vectorized leg. In both of these cases, a new spacecraft trajectory is computed and written to the Spacecraft Ephemeris File.

Table 1: Summary of AutoNav Commands

Command Name	Description	Arguments	Usage	Time required
Nav_Do_OD	Perform Orbit Determination	none	1/week	10-100mins
Nav_Do_TCM	Execute a TCM	duration	1/week	1.5-24hours
Nav_IPS_Off_Mes*	Notify Nav of a forced "engine off"	none	1/week*	1 second
Nav_Man_Plan	Perform Maneuver Planning	none	1/week	10-200mins
Nav_Photo_Op	Perform a nav picture taking and processing session, edit and store data.	duration	1/week	1.5-8hours
Nav_Reset*	Stop all Navexec state machines	none	Seldom*	1 second
Nav_Set_IPS	Start a Mission Burn	none	1/week	5 mins
Nav_Start_Encntr	Start an encounter sequence	seq. ID	4/encounter	1 minute
Nav_Update_IPS	Update the thrust vector during a mission burn	none	2/day	1 minute
Nav_Change_Mode	Change an AutoNav operating mode	Data vectors	2/month	5 secs
Nav_Data_Downlnk	Downlink a Nav file	file ID	2/month	20 secs
Nav_Data_Update	Update a Navigation file	file ID	2/month	20 secs
Nav_IPS_Press	Pressurize the main engine	none	1/week	1-30mins
Nav_ACM_Infoturn	Optional desired pointing of the spacecraft after a nav event	"turnspec"	1/week	5 secs
Nav_BBC_Deadband	Optional desired deadband of the spacecraft after a nav event	deadband	1/week	5 secs
*Contingency or emergency back-up command				

Nav_Photo_Op: Causes AutoNav to 1) cycle through its list of candidate "beacon" asteroids, taking each in turn, 2) for each, query ACS for the turn specifications to take the MICAS boresight to that attitude, 3) before turning, determine that there is sufficient time to turn to target, take the required pictures, and turn back to the desired terminal attitude, 4) if there is sufficient time, turn the spacecraft, 5) begin taking a sequence of pictures, sending each when complete to the AutoNav picture processing element, 6) as each picture is processed, write its reduced data (asteroid pixel, line, pointing values) to the OPNAV file, as well as edited picture elements, 6) cycle to the next asteroid target, via steps 2-5, 7) when the list of candidates is exhausted, or the available time (as communicated in the command argument list) is exhausted, the spacecraft is commanded to turn to the terminal attitude, and 8) the contents of the OPNAV file are filtered for bad data, and the results placed in the OD file, whereupon the OPNAV file is optionally scheduled for downlink and deletion.

<u>*Other AutoNav Commands*</u>

Nav_Reset: Causes any of the three AutoNav state machines, PhotoOP, MissionBurn, or TCM to reset to the off state, if they are active. *Nav_IPS_Off_Mes:* The ground uses this command to inform AutoNav that IPS thrust has been forced off. This will terminate the Mission Burn State Machine, if active. *Nav_Set_IPS:* Causes the initiation of a Mission Burn, by 1) reading the Maneuver File, and determining that a Mission Burn begins within a specified time, 2) querying ACS for the specifications of the turn to the burn attitude, and 3) building and starting a sequence to start at the mandated burn start time (or immediately, if the "Set" command has occurred within a burn segment) that turns the spacecraft, and commands IPS to go to a thrusting state, at the appropriate throttle level, and for the specified duration. *Nav_Start_Encntr:* Causes AutoNav to build and start a sequence that in turn starts the specified sequence at the requested encounter relative time (see RSEN description below). This command is only operable while RSEN is active.

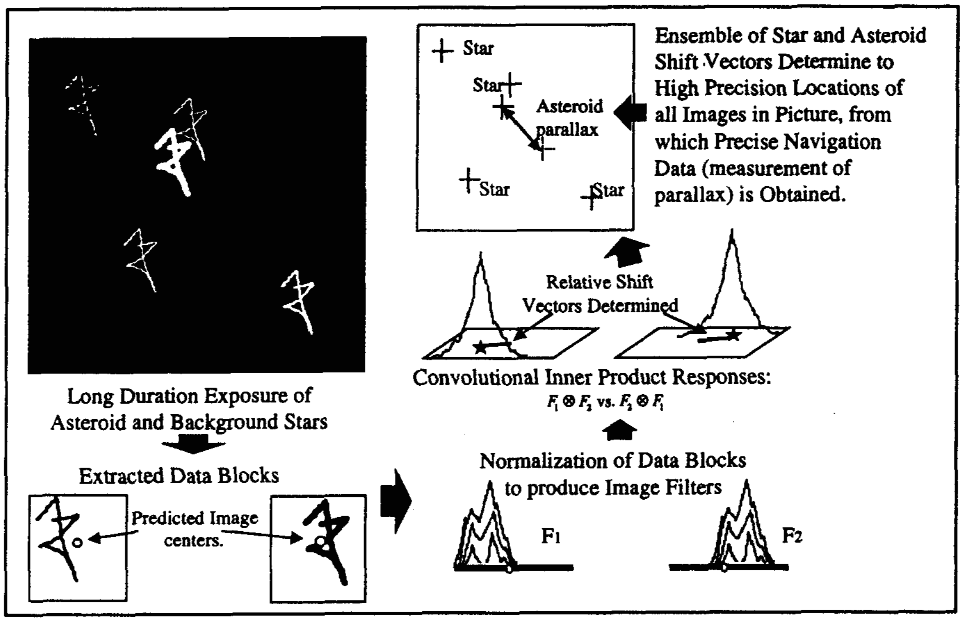
Nav_Update_IPS: During a Mission Burn (i.e. after a Set_IPS command) this command will cause Nav to update the current burn direction according to the Maneuver File. *Nav_Change_Mode:* Updates various control-mode flags, and constant settings in AutoNav. The flags and variables so set are those that need to be changed frequently, or due to changes in spacecraft state or mission phase. Other, more stable, parameters are kept in the parameter files. *Nav_Data_Downlnk:* Causes AutoNav to downlink a specified AutoNav data file. *Nav_Data_Update:* Causes AutoNav to accept a specified AutoNav data file as replacement for an existing file. The AutoNav file of filenames is updated in this process. *Nav_IPS_Press:* Causes AutoNav to command the IPS to pressurize the Xenon chambers in preparation of thrusting at the throttle level determined from the Maneuver File. *Nav_ACM_Infoturn:* Allows the ground to inform AutoNav what the desired ACS turn specification is for the desired terminal attitude after a PhotoOp or TCM. *Nav_BBC_Deadband:* Allows the ground to inform AutoNav what the desired deadband is after a PhotoOp or TCM.

"Uncommanded" AutoNav Functions

Reduced State Encounter Navigation (RSEN), and Encounter Sequence Activation: The encounter navigation activity is performed by this AutoNav subsystem. RSEN is enabled by a Nav_Change_Mode Command, whereupon the most recent estimated spacecraft state and covariance are mapped to the current time. RSEN is then activated by the receipt of an APS picture. When an APS picture is received, the state and covariance are mapped to the picture time by a simple linear motion propagation, the centroid of the target is located in the frame, differenced with a predict to obtain a residual, and a Kalman filtered estimate of spacecraft position is made. Then, the cartesian spacecraft state is converted into "B-plane" coordinates, including linearized time of flight to closest-approach; the time-of-flight information is made available to other AutoNav subsystems. This process continues with subsequent pictures, with RSEN "boot-strapping" states from picture time to picture time. When AutoNav receives a Nav_Start_Encntr command - wherein Nav is asked to start an encounter sequence at a specific time - the time of closest approach previously computed by RSEN is compared with the current time, and an absolutely timed sequence is built to start the desired sequence at the appropriate time.

Non-Grav History Accumulation: AutoNav must keep a continuous record of propulsive events by RCS and IPS onboard the spacecraft for purposes of accurately integrating the flightpath of the spacecraft. In this effort AutoNav is aided by the ACS and IPS software susbsystems, which report periodically accumulated delta-v (in the case of ACS) or impulse (in the case of IPS). The periodicity of reporting varies for ACS, since this system buffers the accumulation, and only reports when a certain threshold is crossed, typically 10mm/sec. For IPS the reporting is every minute. AutoNav further buffers this data under parametric control, writing "permanent" records in EEPROM when accumulated ACS delta-v or IPS vector impulse cross internal AutoNav thresholds

Figure 2: Multiple Cross Correlation of Asteroid and Stars



Ephemeris Services: Is the highest priority AutoNav task, and is required to give ephemeris information to ACS as often as on one second intervals, under some rare circumstances, but nominally is queried every few minutes. The ephemeris reads the ephemeris files of the spacecraft, the beacon asteroids, and the major planets. All of these files have Chebyshev polynomial representations of the orbital states, with velocities computed. All states are in Earth-Mean-Equator-2000 coordinates, as are the directions on the Star Catalog. Ephemeris Services provide ephemeris data to the internal AutoNav functions as well.

Core Algorithm Descriptions

Multiple Cross Correlation: Figure 2 shows a diagrammatic representation of the algorithm that forms the basis of the cruise image processing in AutoNav. The underlying assumption of the algorithm is that long exposures will be necessary to image dim objects, and thus, because of ambient motions of the spacecraft due to attitude maintenance by the ACS, the images of stars and target will be smeared, often in complicated patterns. These patterns, called "glyphs" will be nearly identical to one another, since the effects of "twisting" deadband motion in the field is small (the attitude maintenance is roughly equivalent in all directions, but "twist" maps to a much smaller effect in the field than the two cross line-of-sight pointing directions.) Based on initial knowledge of pointing of the spacecraft (as provided by ACS) and predictions of the relative locations of the objects in the field of view (based on the target ephemeris and the star catalog) segments of the pictures are extracted, normalized and these become templates or "filters". Filters for each object are used to locate each of the other objects. The "location" is accomplished through the convolutional inner-product of filter with data. Once all of the objects are located relative to one another (and these data filtered for bad or weak signal), a least squares estimate is made of the relative offset of the objects relative to one another. A complete description of this algorithm is given in Ref. 2, as it was used for the *Galileo* Gaspra encounter.

Orbit Determination : Figures 3a,b,c give an outline of Orbit Determination and related algorithms as used by AutoNav. There are several crucial elements to the Orbit determination function: 1) the numerical integration of the spacecraft trajectory (Figure 3a), 2) the dynamic models of the gravitational and non-gravitational perturbations that drive that integration (Figure 3a), 3) the generation *of,* and the mapping of the covariance in time *with* the state transition matrix (Figure 3b), and 4) the formation of the data filter itself (Figure 3c). As noted earlier, the OD filter used is a Kalman batch-sequential least-squares filter. A typical data arc is about a month long, with four 1-week batches that correspond to the typical one Photo-Op event per week. The estimated parameters for a given solution include the position and velocity at the beginning of the data arc, a constant acceleration 3-vector which applies for the duration of the arc, and IPS thrust scale factors that are stochastic parameters for each week. The latter parameters are in force only while there is an

Figure 3a,b,c: Spacecraft Integration Equations of Motion, and Derivation of AutoNav OD Kalman Filter

Dynamical equations of motion

- **I ncludes central body acceleration, 3rd body perturbations from other planets, solar radiation pressure, thrust from the ion engines, and miscellaneous accelerations**
- **2 nd order differential equation modeled as two 1st order differential equations**

$$\dot{\mathbf{r}} = \mathbf{v}$$

$$\dot{\mathbf{v}} = -\frac{\mu_s}{r^3}\mathbf{r} + \sum_{i=1}^{n_p} \mu_i \left[\frac{\mathbf{r}_{ri}}{r_{ri}^3} - \frac{\mathbf{r}_{pi}}{r_{pi}^3}\right] + \frac{AG}{mr^3}\mathbf{r} + \frac{k}{m}\mathbf{T} + \mathbf{a}$$

where

$\mathbf{r}$ = the heliocentric cartesian position vector of the spacecraft
$\mathbf{v}$ = the heliocentric cartesian velocity vector of the spacecraft
$\mathbf{r}_{pi}$ = the heliocentric cartesian position vector of the *i*th perturbing planetary body
$\mathbf{r}_{ri}$ = the position of the spacecraft relative to the *i*th perturbing body
μ_s = the gravitational constant of the sun
μ_i = the gravitational constant of the i_{th} perturbing planet
n_p = the number of perturbing planets
A = the cross-sectional area of the spacecraft
G = the solar flux constant
T = the thrust vector from the ion engine
k = the thrust scale factor
m = the spacecraft mass
$\mathbf{a}$ = miscellaneous accelerations acting on the spacecraft

Given $\mathbf{q}^*$, the nominal trajectory parameters, as

$$\mathbf{q}^* = [\mathbf{r} \quad \mathbf{v} \quad k \quad \mathbf{a}]$$

Filter estimates corrections, $\mathbf{q}$, to nominal trajectory parameters

$$\mathbf{q}(t) = [\Delta x \quad \Delta y \quad \Delta z \quad \Delta\dot{x} \quad \Delta\dot{y} \quad \Delta\dot{z} \quad \Delta k \quad \Delta a_x \quad \Delta a_y \quad \Delta a_z]$$

The correction at time t is a linear mapping of the correction from time t_0

$$\mathbf{q}(t) = \Phi\mathbf{q}(t_0)$$

where Φ, the state transition matrix, is defined as

$$\Phi(t) = \frac{\partial\mathbf{q}^*(t)}{\partial\mathbf{q}^*(t_0)}$$

The partial derivatives of the observed pixel and line locations, p, l, with respect to the state, at time t is

$$\mathbf{H}(t) = \begin{bmatrix} \partial p/\partial \mathbf{r} & 0_{1x7} \\ \partial l/\partial \mathbf{r} & 0_{1x7} \end{bmatrix}$$

This can be mapped back to the epoch, t_0, via the state transition matrix

$$\tilde{\mathbf{H}}(t_0) = \mathbf{H}(t)\Phi$$

The minimum variance least squares solution to the epoch state corrections is

$$\hat{\mathbf{q}} = \left[\mathbf{P}_0 + \tilde{\mathbf{H}}^T\mathbf{W}\tilde{\mathbf{H}}\right]^{-1}\tilde{\mathbf{H}}^T\mathbf{W}\mathbf{Y}$$

where

$\mathbf{P}_0$ = the a-priori covariance of the state parameters
$\mathbf{W}$ = the weighting values of the pixel and line observables
$\mathbf{Y}$ = the residual vector between the observed pixel/line locations and their predicted values

Mission Burn in progress during that portion of the arc. (Please see Reference 5 for complete details).

IPS Mission Burn Targeting: The process for retargeting the spacecraft trajectory during a Mission Burn is shown in Figure 4. This is an iterative application of a linear estimation of corrections to the direction of burn of an individual element of the multi-element Mission Burn, and the duration of the final element. Since iterative, the overall algorithm is non-linear. The algorithm will automatically decide how many segments to include in the solution, starting with a minimum acceptable number, and increasing the number as necessary to gain sufficient control authority to achieve convergence, i.e. putting the spacecraft on target. (See Reference 6 for more details).

Figure 4: Adjusting a Low Thrust Burn Arc

X_e = Target encounter conditions $(B \cdot R, B \cdot T, tof)$

ΔX_e = Deviations from desired encounter

α_i = Rt. ascention of thrust segment i

δ_i = Declination of thrust segment i

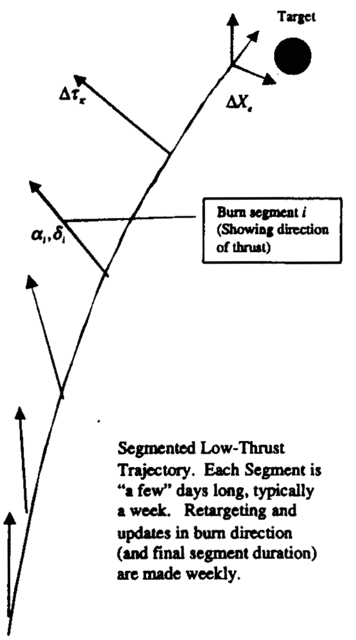
τ_κ = Burn duration of final segment κ

$\Delta X_e = K \Delta s$, where...

$$K^T = \begin{bmatrix} (\partial X_e / \partial \alpha_{k_1}) \\ (\partial X_e / \partial \delta_{k_1}) \\ (\partial X_e / \partial \alpha_{k_1+1}) \\ (\partial X_e / \partial \delta_{k_1+1}) \\ \vdots \\ (\partial X_e / \partial \alpha_{k_2}) \\ (\partial X_e / \partial \delta_{k_2}) \\ (\partial X_e / \partial \tau_\kappa) \end{bmatrix}, \text{ and... } \Delta s = \begin{bmatrix} \Delta \alpha_{k_1} \\ \Delta \delta_{k_1} \\ \Delta \alpha_{k_1+1} \\ \Delta \delta_{k_1+1} \\ \vdots \\ \Delta \alpha_{k_2} \\ \Delta \delta_{k_2} \\ \Delta \tau_\kappa \end{bmatrix}$$

Then, by the calculus of variations,

$$\Delta s = K^T (KK^T)^{-1} \Delta X_e$$



It is important to note that the spacecraft is given a "converged" trajectory initially. This trajectory has been "discovered" and reasonably converged initially with an algorithm known as "differential inclusion" (Reference 6) and uplinked to the spacecraft. Then, within well regulated limits, the maneuver planner is allowed to adjust this trajectory to keep the spacecraft targeted.

Interdependencies with other Onoard Systems

Attitude Control System (ACS): AutoNav has mission-critical interfaces with ACS. Basic spacecraft health is dependent upon Nav providing ACS with the locations of the spacecraft and requested target bodies. Without this information, the spacecraft will be forced (under certain circumstances) into safing. In order to accomplish its autonomous activities, Nav communicates with ACS in several ways. Though not explicitly called out as a technology demonstration of *DS1*, the design and implementation of the *DS1* ACS system contain a number of important technological advances. These include the operation of the IPS, attitude maintenance and turns with highly constrained attitudes, and autonomous turn planning for AutoNav. When NavExec desires to change the attitude of the spacecraft, it queries ACS for the particulars of the turn between the assumed beginning attitude and the desired attitude. ACS will inform NavExec 1) whether the turn is possible at all, 2) whether it violates (or nearly violates) any pointing constraints, 3) how long the turn will take. Armed with this information, NavExec decides whether to proceed.

During the course of its autonomous work, AutoNav has occasional need to alter the operational state of ACS. These changes include changing from normal RCS (Reaction Control System) mode to TVC (Thrust Vector Control) mode when operating the IPS is required. The mode which controls the pairs of thrusters used to turn the spacecraft must be set to allow for "slow" deadband maintenance during picture-taking is also altered. For most of the spacecraft actions AutoNav commands, the attitude control deadband itself must be changed to suit the activity. In addition, the ground generated sequence must set the family of constraints which proscribe areas on the spacecraft from sun-illumination before certain AutoNav events.

As stated earlier, ACS periodically queries NavRT for ephemeris information. These queries always include a request for the spacecraft position, and a request for the position of the body (if any) toward which the spacecraft is currently pointing. ACS also records all propulsive activity from the RCS, and computes a net translational change in velocity (delta-v). When the value of this delta-v is greater than a predetermined value, a message containing the accumulation is sent to AutoNav, and, after further buffering, these quantities are recorded on the AutoNav NonGrav History file.

Because of the sun-illumination constraints (and geometric constraints involving keeping the solar panels focused on the sun), it is impossible to point the spacecraft in certain directions. If it is necessary to accomplish a TCM in one of these directions, it is necessary to break the vector up into two components which are allowed. ACS provides a service wherein AutoNav requests a delta-v direction, and ACS responds with one or two allowed directions for burning the engines. Upon receipt of this information, AutoNav recomputes the magnitudes of the burn elements if it in fact has been so "vectorized". When the final values of the TCM have been computed, Nav turns the spacecraft (through interaction with ACS) and asks for either an RCS delta-v, or causes the IPS to burn for a specified time.

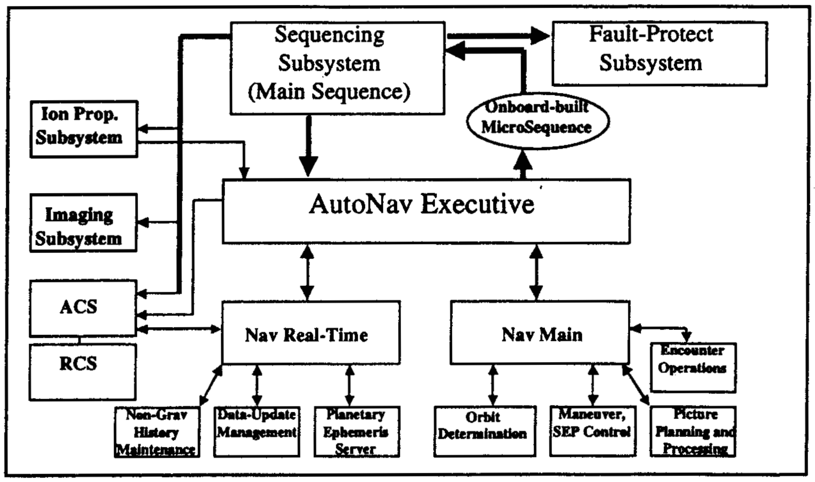
Ion Propulsion System (IPS): AutoNav has responsibility to perform basic operation of the IPS during Mission burns and TCMs that use IPS. Additionally, IPS is responsible to report to Nav the progress of any IPS thrusting. Nav commands IPS through directives to pressurize at a given thrust level, ignite the engine, stop and safe the engine. IPS in turn gives reports of the accumulated impulse over a one minute period, and reports when the specified duration of the burn has been achieved. When this last message is received, Nav commands the engine to shut down. Accumulated IPS impulse is recorded on the NonGrav History file.

Fault Protection: One of the fundamental guidelines in the design of the AutoNav system was to minimize the possible amount of trouble that the system could cause other systems or the spacecraft overall. AutoNav to a very large degree attempts to trap all of its possible errors internally, and exit the faulty function in a manner that to the external system looks "normal." As a result, there were no explicit connections to the FP system. It was additionally felt that none of the types of internal Nav failures mentioned above warranted notice by FP, even in a monitoring sense. Furthermore, the general use of the sequencing system for most commanding that involved actual spacecraft actions meant that AutoNav requests for action were covered by the usual Fault Protection provided by any sequence. There is one indirect method by which FP can detect an AutoNav failure. During certain fault recovery modes when ACS does not receive ephemeris data from AutoNav, it complains to FP, which will variously, depending upon circumstances, merely note, or take the spacecraft to a higher level of fault-state. As part of a safing event, FP will run scripts that set the AutoNav Modes into "stand-by" states wherein no attempts will be made to alter EEPROM files, including the Non-Grav History File.

AutoNav Software System

The AutoNav software is shown schematically in Figure 2. The AutoNav system is composed of three principle parts, the Nav Executive, Nav Main, and Nav Real-Time (NavRT). These communicate with each other and other subsystems through the underlying system messaging facility. Much of the commanding by AutoNav is through the sequencing subsystem, as will be discussed below.

Figure 5 The AutoNav Software System and Interacting System Software



Nav Executive: NavExec is AutoNav's director of spacecraft activities. It receives messages from other s/c subsystems and sends command directives, either through the onboard sequence machine or through direct messages to other subsystems. When using the sequence subsystem (sequence engine), NavExec will build small sequences, and "launch" them. When NavExec needs an activity to occur immediately, for example to turn the spacecraft to a desired burn attitude, it will build a relative time sequence which the sequence engine initiates at once. Alternatively, when NavExec needs to insure that an event begins exactly at a certain time, it will build and initiate an absolute timed sequence, for example to cause the main engine to ignite for a TCM. NavExec contains three main state machines, for Photo-Ops, TCMs and for Mission Burns. These machines are mutually exclusive, the activities involved being clearly incompatible.

Nav Real-Time: NavRT is the subsystem of AutoNav that provides critical onboard ephemeris information to other onboard subsystems, but principally to ACS. NavRT operates at a much higher priority level in the flight-software than the other AutoNav components, due to the need to respond to sometimes frequent and time-critical ACS requests. NavRT also accomplishes file updates, involving ephemeris related files, by insuring that changes in files are completed in a way as to not jeopardize ACS ephemeris queries.

Nav Main: or just plain "Nav", is the central computing element of AutoNav. Requests for activity that involve large amounts of computing are directed to Nav by NavExec, or go to Nav directly through the command subsystem. These functions include picture processing requests from NavExec, Do-OD and ManPlan commands from ground commands. There are several important sub-functions of Nav: trajectory integration, which includes dynamic modeling of gravitational and non-gravitational forces acting on the spacecraft, data filtering, including a U-D factorized batch-sequential filter, and trajectory update computation, based on an iterative linear minimum-norm solution for changes to the IPS thrust profile to reduce projected targeting errors.

The Operations Plan for DS1

Figure 6 gives a detailed overview of planned DS1 operations which involved AutoNav. This version of the plan, prepared by the Mission Design Team, shows only a subset of the intense activities that were planned and executed in the initial months of DS1 operations. Since the principal goal of DS1 was to validate its suite of eleven technologies, and these were flying aboard a relatively inexpensive vehicle, quickly developed, it was imperative to perform as many validation experiments as early in the mission as possible. The figure gives a very good indication of the compressed nature of the schedule. First use of the ion engine for mission critical thrusting was to beinn within 2 weeks of launch, as was use of AutoNav. A fault in the IPS caused several weeks of delay, and major overhaul of the schedule. First images taken with MICAS revealed enormous scattered light problems, to be discussed at length below. Within 8 months of launch, encounter sequences needed to be built, and tested in an onboard rehearsal, including all of the TCM and encounter AutoNav capabilities. This was a very challenging schedule. The navigation results from this plan will be discussed at length below.

AutoNav Dependence on the *DS1* Imaging System

For *DS1*, the camera, like AutoNav was another technology being demonstrated. MICAS, the Miniature Imaging Camera And Spectrometer has two visual channels, a somewhat conventional CCD (Charge Coupled Device) detector and a much smaller APS (Active Pixel Sensor). Both of these channels are continuous read-out sensors, and are shutterless. The ability to take high quality astrometric images of small asteroids and image a bright inner solar-system target against a field of stars presents stringent requirements on a visual detector. Eight months before launch, it was discovered that the CCD channel had a severe limitation when imaging bright objects (objects as bright as the first two expected targets). When an object of a typical asteroid brightness subtended more than 100 pixels (+/-50), severe charge bleed appeared in the picture due to the inability of the CCD read-out to cope with the continuing photon flux during the read-out. Because of this limitation, it was believed that the CCD channel would be unusable during the last few minutes of approach. Figure 7 shows an example of the phenomena, taken before launch.

Figure 6: DS1 AutoNav and Related Operations Schedule

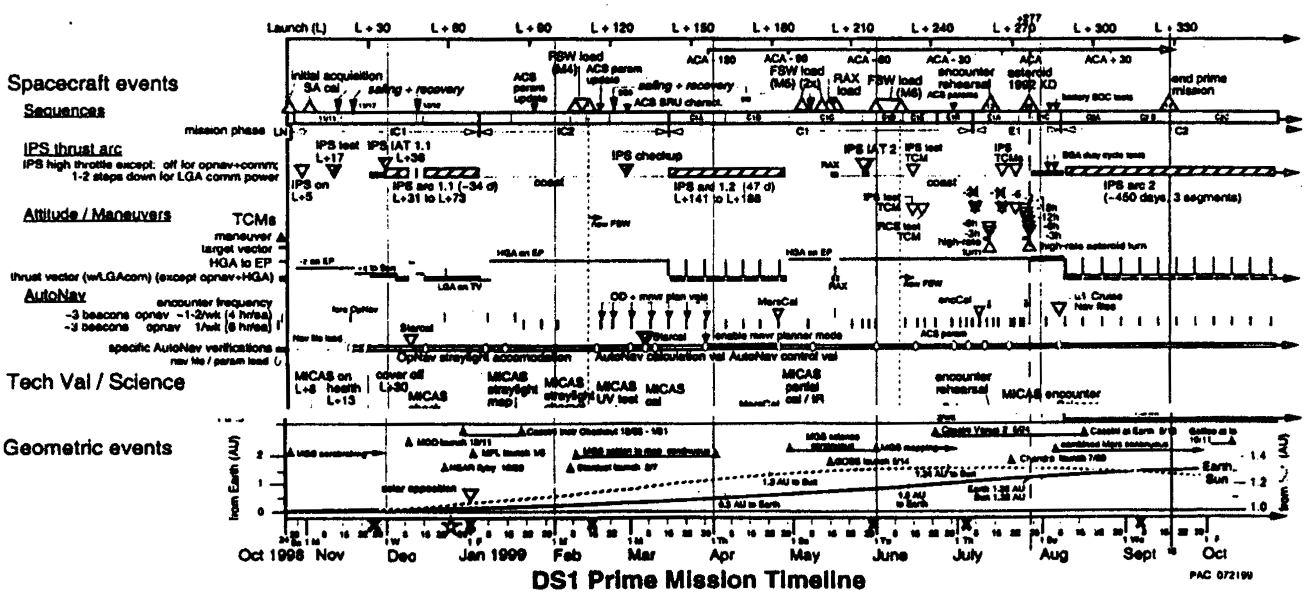


Figure 7: MICAS Extended Bright Image Charge Bleed

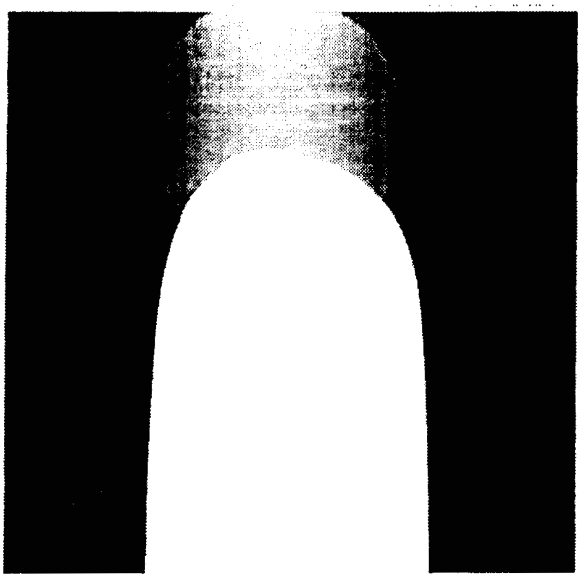
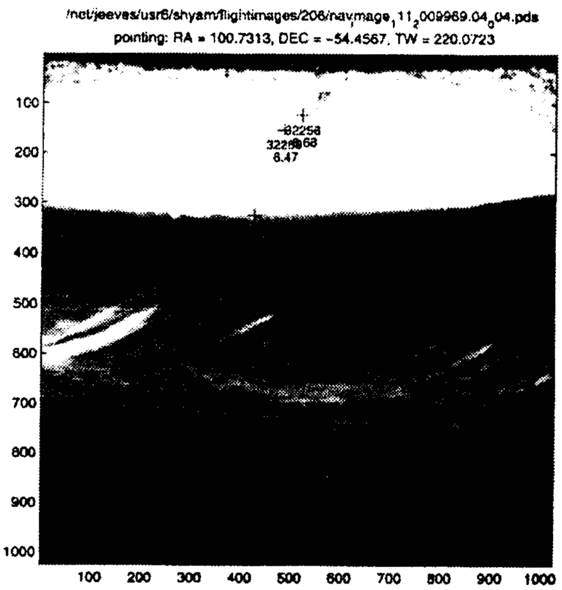
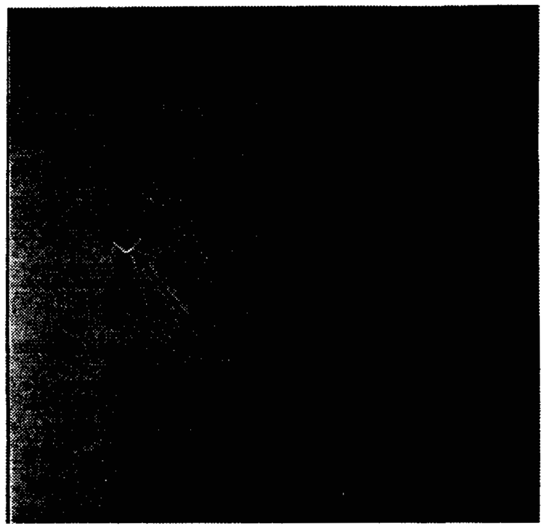


Figure 8: MICAS "Low Solar Cone Angle" Scattered Light Picture



As a result of this problem, the less capable APS channel was used by AutoNav on approach. In partial compensation, the read-out time required for the APS was much shorter than for the CCD, 2 seconds vs. 20 seconds. This shorter time allowed a higer frequency of image taking, which somewhat balanced the much smaller field-of-view of that channel. A much smaller field greatly limited the robustness of AutoNav to ACS pointing or ephemeris errors. At the first use of MICAS it was apparent that there were substantial light scattering problems around and in the camera (Ref. 4). Depending upon the sun-relative geometry, the CCD would saturate (achieve maximum measurable charge) in as little as 5 seconds of exposure. In view of the fact that the original feasibility analysis of AutoNav called for exposures as long as 200 seconds, this clearly represented a reduction in capability by limiting usable geometries and targets. Figures 7 and 8 show two examples of the scattered light effect in roughly normal-to-sun and anti-sun geometries. Despite these severe difficulties however, asteroids were still visible. In Figure 9 are indicated asteroid Vesta (7th Magnitude in this image) and a star (8th Magnitude). With substantial upgrades to the AutoNav software, other dimmer asteroids and stars were eventually obtainable and autonomously processable, as will be discussed below. A third difficulty with the camera is a highly non-linear response curve (Figure 10.) The net effect of this

Figure 9: MICAS "High Solar Cone Angle" Scattered Light Picture

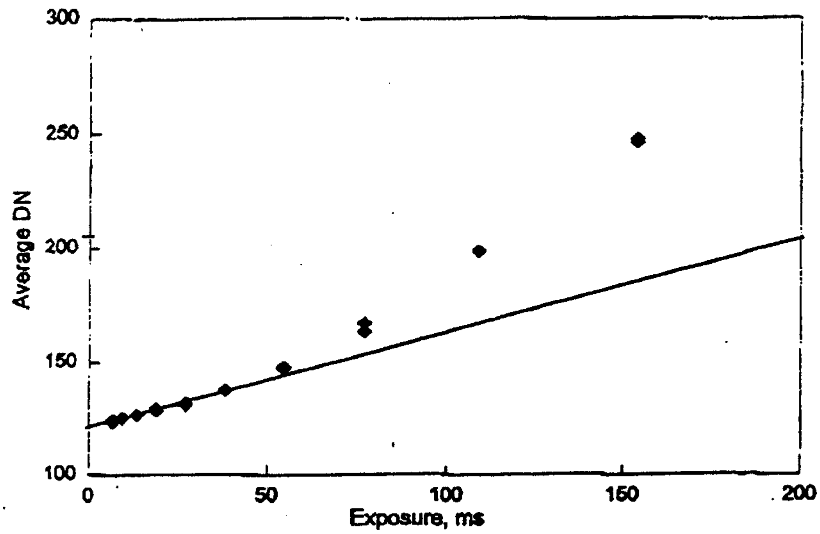


electronics fault is for low flux signals to be non-linearly attenuated. This effect is much more severe in the APS, and largely accounted for abnormally low throughput at the Braille encounter. Yet another substantial difficulty for AutoNav arose due to light-attenuating scratches in the optics-chain over a substantial portion of the CCD center of field-of-view. These blemishes are partially shown as the dark streaks and patterns in the center and center top of Figure 9.

AutoNav Technology Validation

The overarching philosophy behind AutoNav operational testing, was to initially ground test every operation of AutoNav under normal and a selection of abnormal circumstances. Once in flight operations, the first few events of a given Nav operation were always tested on various testbeds thoroughly. Only after several successful operations under this closely simulated test restriction were the autonomous systems allowed to operate without a very well tested predict of the expected outcome.

Figure 10: MICAS APS Channel Non-Linear Signal Response

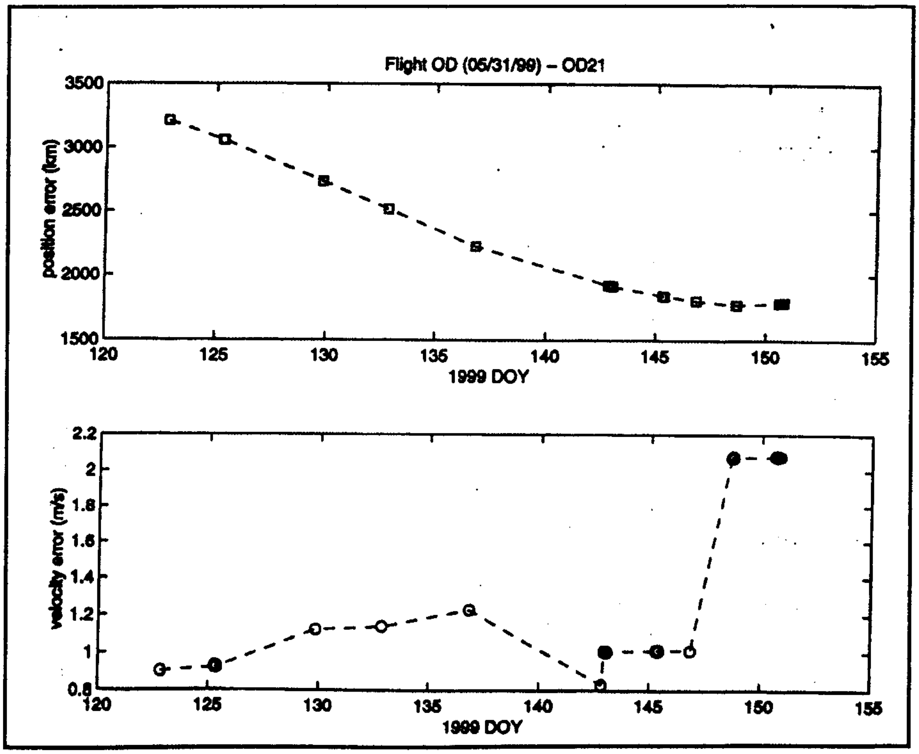


The principal difficulty in this strategy was the early almost complete lack of predictability of the behavior of the scattered light and leakage within the MICAS camera. As is discussed in the body of the report, this problem caused general failure of the image-processing algorithms, depriving subsequent functions of data, altering the expected behavior of the AutoNav sessions. In no case however, was this inability to predict considered to be, nor did it at any time prove to be a hazard.

The concept of an autonomous optical navigation system was proved early in the mission development phase using a MATLAB® simulation of a ballistic mission to an asteroid. This demonstration simulated pictures taken in flight by such a mission, processed those pictures and used the reduced data in a orbit-determination estimation process. Subsequently, maneuvers were computed to control accumulated errors in the simulated orbit due to OD errors, non-gravitational model errors and perturbations. Finally, the encounter was simulated, with late tracking and orbit updates of the target. Results from this simulation gave strong indication that orbit quality of better than 500km and 0.5 m/s was possible during the cruise phase, as well as delivery at the target to better than 10km (Reference 5).

As the actual flight system began to develop, tests were on-going, covering a wide range of expected mission operating conditions. Early in this process, the decision was made to make *DS1* a low-thrust mission, requiring a substantial increase in the complexity of AutoNav. Extensive new theoretical development and test was required as a result (Ref. 6). Of a large number of missions considered and partially evaluated, a mission to asteroid McAuliffe, then Mars, followed by a flyby of comet West-Kahotek-Ikamoura was settled upon, and extensively evaluated. The extensive cruise phases were simulated and OD performance evaluated, and the ability of the maneuver planner to keep the spacecraft on course was robustly demonstrated. This mission was subsequently replaced by a new design, targeting 1992KD, Wilson-Harrington, and comet Borelly mission, due to a required launch delay. (As a side note, as of March 2000, the DS1 mission has changed again. In November of 1999, the star-tracker failed. After extensive efforts to diagnose and correct the problem proved fruitless, a major redesign of the onboard ACS system was undertaken, which, using MICAS images, and redesigned components of AutoNav, will commence a new method of attitude control in May of 2000. Unfortunately, the ensuing time lost precludes achieving the more than the Borelly fly-by.)

Figure 11: Flight vs. Ground Orbit Determination, May 31, 1999



Each of the elements of AutoNav went through stand-alone tests and extensive system tests as part of the delivery process of each new version of the software. The system tests covered various mission phases, and all of the interactions and functions of Nav. Additionally, AutoNav systems, particularly the ephemeris services, were required for all other system tests, leading implicitly to additional Nav verification. None of these tests gave performance and capability results in conflict with the prototype demonstration phase.

Upon the first invocation of the higher AutoNav functions in flight, in November of 1998, it was obvious that pre-flight performance estimates would not be met; this was almost entirely due to the problems encountered with MICAS. Because of the scattered light leakage problems it was impossible to successfully acquire navigational data onboard before extensive AutoNav flight-software modifications were performed. However even ground processing of the onboard acquired images revealed problems, keeping the performance of the system (as demonstrated on the ground) above 5000km and 2m/s (Figure 12). Nevertheless, all other subsystems of AutoNav, including autonomous picture-taking planning execution, IPS mission burn planning and execution, orbit determination with ground-seeded data performed well.

By June, 1999, all modifications had been made to the cruise AutoNav system, including image processing changes to deal with the scattered light-leakage problems, and severe geometric distortions observed in the camera field. With these changes and calibrations onboard, the performance of the onboard cruise navigation on several occasions met the originally forecast performance values of better than 250km and 0.5m/s. However, due to the continuing uncertainty of the geometric distortions, this could not be continuously maintained autonomously onboard, but could through hand editing of data on the ground and subsequent upload of the edited data sets; see Figure 12. But this performance was sufficient to continue with the validation schedule and use AutoNav for approach to Braille.

On July 13 1999, just 16 days before closest approach a full onboard "dress rehearsal" of the encounter was performed, with AutoNav simulating the encounter with a "pseudo-Braille," autonomously computing and executing one of the two approach TCM's, and delivering the spacecraft within 2.5 km of it's fictional target. AutoNav also started encounter sequences as close as 5 seconds to the nominal encounter-relative start time. All subsystems executed perfectly, and the *DS1* team as well as AutoNav were primed and ready for the actual approach.

Figure 13 shows a series of solutions from the onboard AutoNav system and from the radio navigation system on the ground used to develop the Encounter–5day TCM. TCM's at –20 and –10 days were cancelled due to the relative stability of both flight and onboard solutions and nearness to the desired target position. These solutions were all made before the initial onboard sighting of Braille, and were based only on *a priori* estimates of the asteroid ephemeris. Two such pre-encounter ephemerides are shown. Also shown is the close agreement of the ground radio and AutoNav solutions. The 7/22 AutoNav Flight Solution was used onboard to compute a TCM labeled "B", virtually identical to the ground computed solution based on "G". The following day, the flight solution had shifted to "F2", and a new encounter-5 day burn was computed, but due to an anomaly with the onboard file system, solution "B" was reverted to. It was felt that this solution left the spacecraft within 200km of the proper target conditions.

After the "-5 day TCM" there were TCM opportunities scheduled at –2d, -1d, -18h, -12h, -6h and –3h. Throughout this time, observations were scheduled to allow AutoNav the opportunity to attempt a detection of Braille. Because of the non-linearity of the imaging

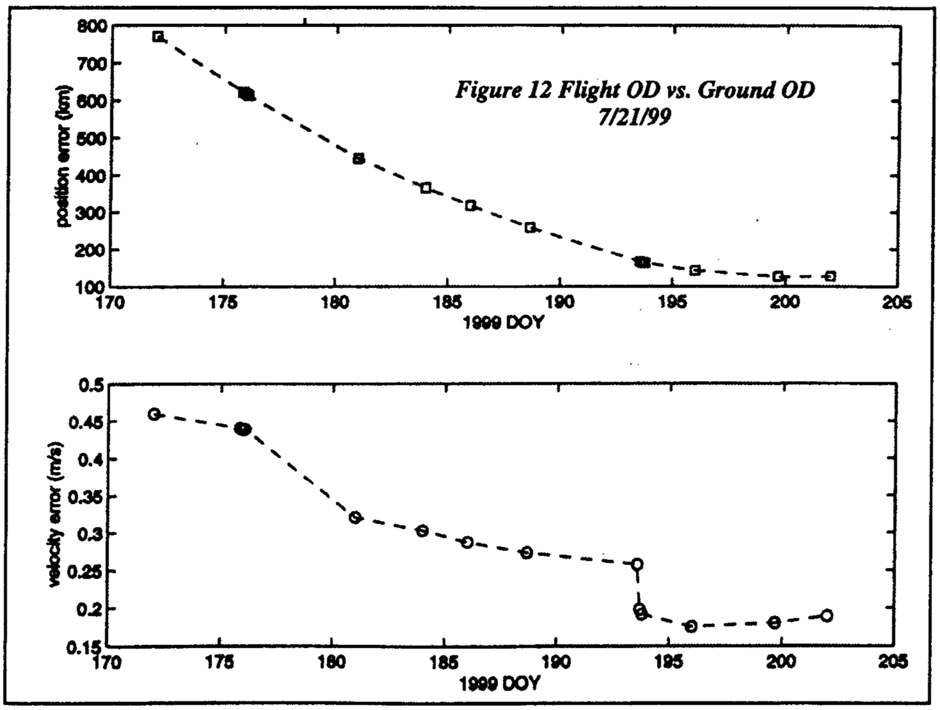


Figure 12 Flight OD vs. Ground OD 7/21/99

system and the inability to take long exposures due to scattered light, the first detection of Braille wasn't made until encounter -2.5 days. This detection was only by ground operators however, due to the faintness of the signal. However it showed a 350km ephemeris shift (approximately 2 sigma). Because of this large shift, and the faintness of the observation, the –2d TCM was cancelled (AutoNav would normally decline to execute a TCM if it was statistically insignificant, but because of the data difficulties associated with the geometric corrections, it was decided to actively prevent this execution). In subsequent observations, the dim "phantom image" turned out to be true, but remained anomalously feeble (due to the previously discussed camera non-linearity). As a result, AutoNav was unable to "lock-on" to Braille prior to the "-1 day" TCM, and 5 hand-located Braille images were used to design this TCM, shown in Figure 14. The –1day TCM was completely nominal, and at –18hours, a photo-session was performed in which AutoNav finally "locked on" to a sufficiently bright Braille image, and was proceeding to compute the –12 hour TCM when a software fault caused the spacecraft to "safe". It took almost 10 hours to recover the spacecraft into normal mode, during which time 3 of the surviving pictures from the photo-session were downlinked and used to compute the -6 hour TCM. This data was uplinked to the spacecraft and it was set on its way for the final 6 hours of autonomous operations.

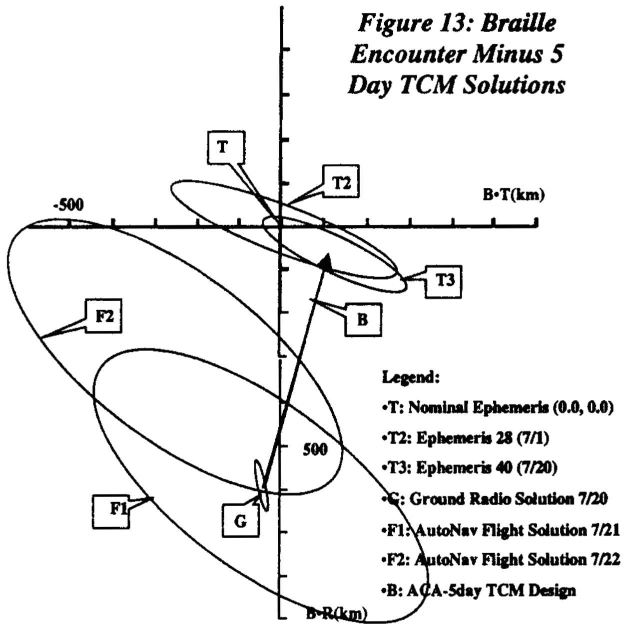


Figure 13: Braille Encounter Minus 5 Day TCM Solutions

Because of the bright-image bleed problems with the CCD, it was necessary for AutoNav to switch detectors to the less capable and less well characterized APS channel shortly before encounter. With nearly all of the science and all of the Nav data scheduled from this sensor within 30 minutes of closest approach, the approach sequence was extremely dependent upon models that described the expected brightness of the approaching target. In the event, the target was far dimmer than expected for at least two reasons. First, the photometric predictions were inaccurate due to inextendability of the assumed models to the encountered geometry, and the lack of allowance for an inopportune presentation of an oblong object to the approaching spacecraft. Second, the APS sensor exhibits extreme non-linearity at low signal, causing a flux dimmed by the first phenomena to have its signal obliterated. As a consequence, no useable signal was received, and effectively, close-approach AutoNav did not operate during the Braille encounter.

Despite the fact that the performance of the system during the Braille flyby was thwarted, it is nevertheless the case that operability and accuracy of the AutoNav close-approach system had been demonstrated in the testbeds, and more importantly in-flight during the rehearsal. This was proved in the real case using the few acquired CCD images of Braille post-encounter. When these were provided to AutoNav, accurate solutions of the spacecraft position were obtained with just 1 CCD image, leading to the unavoidable conclusion, that had this detector been used, instead of the APS, the encounter would likely have been very successful. Fig. 15 shows the B-plane results of this analysis. Figure 16 shows one of the post-encounter APS science images taken 15 minutes after closest approach (Braille is the dim smudge center-left). Despite the fact that Braille appeared several times brighter outbound than inbound, this signal is barely detectable, strongly indicating that the approach APS images available to AutoNav had no discernable images.

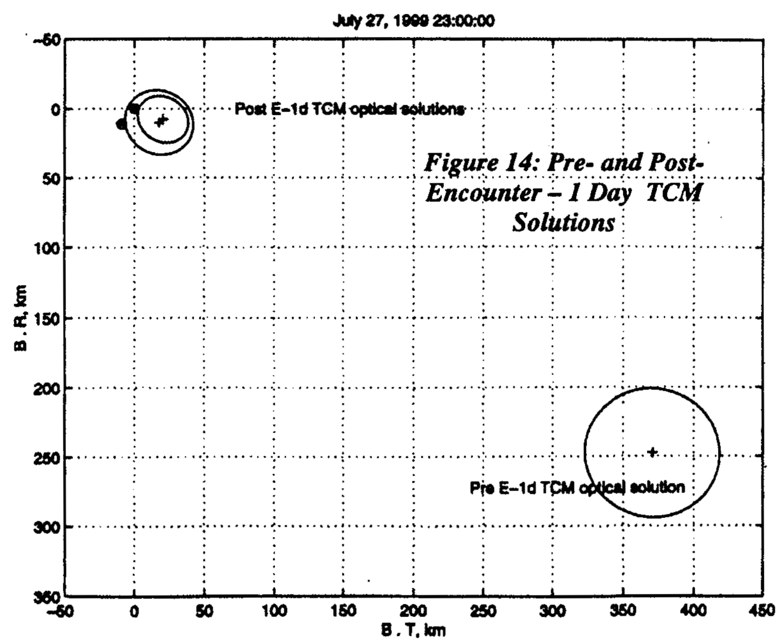


Figure 14: Pre- and Post-Encounter – 1 Day TCM Solutions

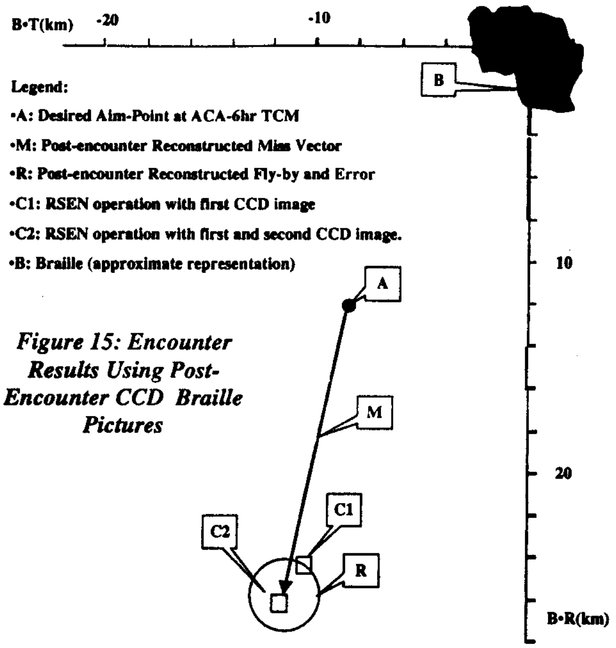
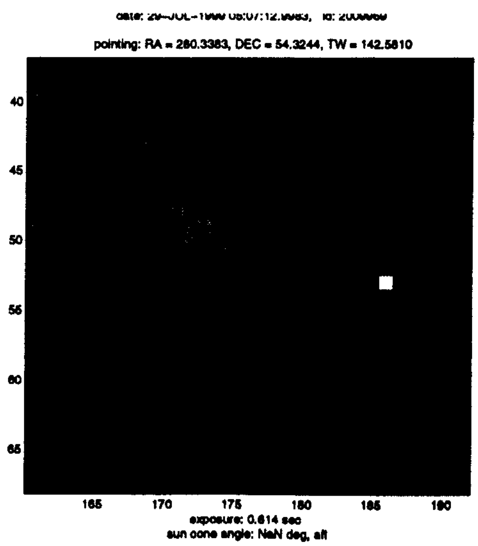


Figure 15: Encounter Results Using Post-Encounter CCD Braille Pictures

Conclusions

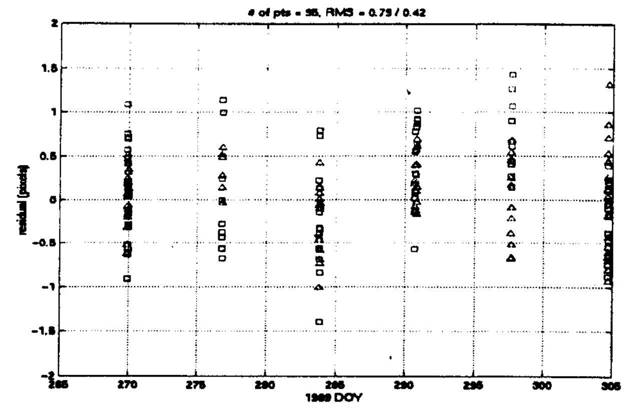
Despite the disappointment of the Braille encounter, the overall success of the DS1 technology validation mission must be rated very highly. In a period of little over 9 months, several advanced and complex technologies were validated while a spacecraft was kept on course for an interplanetary encounter, and achieved that encounter. AutoNav shared fully in that success, and perhaps the best measure of that occurred in the two months immediately following Braille encounter. Then, the *DS1* Navigation team enjoyed the advantages of its work, as the system was invoked and allowed to navigate the spacecraft without intervention. These results, optical residuals from several dozen asteroid observations, are shown in Figure 17. These represent excellent results by any measure.

Figure 16: Post-Encounter APS Image of Braille



Interplanetary navigation autonomy was achieved: *DS1* determined her own course while the Nav team got a well deserved vacation!

Figure 17: Post-Braille AutoNav Data Arc and Residuals



Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1) J. E. Riedel, et al "Optical Navigation During the Voyager Neptune Encounter", AIAA paper 90-2877-CP, AIAA/AAS Astrodynamics Conference, Portland Oregon, August 1990.
2) R. M. Vaughan, J. E. Riedel, R. P. Davis, W. M. Owen, S. P. Synnott, "Optical Navigation for the Galileo Gaspra Encounter," AIAA paper 92-4522, AIAA/AAS Astrodynamics Conference, Hilton Head, South Carolina, August 1992.
3) J. E. Riedel, et al, "Navigation for the New Millennium: Autonomous Navigation for *Deep Space 1*", SP-403, 12th International Symposium on 'Space Flight Dynamics', ESOC, Darmstadt, Germany, June 1997.
4) R. J. Koshel, G. Peterson, "Stray Light Analysis of MICAS," BRO Report #3580, September 1999, Breault Research Organization, Inc, Tucson Arizona.
5) S. Bhaskaran, et al., "Orbit Determination Performance Evaluation of the *Deep Space 1* Autonomous Navigation System", AAS 98-193, AAS/AIAA Space Flight Mechanics Meeting, Monterey, California, February 1998.
6) S. D. Desai, et al. "The *DS1* Autonomous Navigation System: Autonomous Control of Low Thrust Propulsion System", AAS Paper 97-3819, AAS Conference, New Orleans, Louisiana, August 1997.
7) Coverstone-Carroll V., Williams, S. N. "Optimal Low Thrust Trajectories Using Differential Inclusion Concepts", AAS/AIAA Spaceflight Mechanics Meeting, Cocoa Beach Florida, USA, Feb. 1994